Lecture 7 - Graphics and Audio subsystems

"(A picture is worth a thousand words.) An interface is worth a thousand pictures." - Ben Shneiderman

- how do game objects get drawn to the screen?
- how do we talk to the GPU?
- how can we abstract away the low-level complex details to simplify our representation of drawing to the screen?

Audio subsystem

- how do we store sound effects and music?
- how can we play these stored audio files?
- how can we implement positional sound?

Last Week Recap

- Physics
- SPSFA (self-peer-supervisor formative assessment) design

Screen Modes

- Full screen (Exclusive)
- Windowed
- Borderless

Rendering Modes

- Immediate
- Retained

Consider the differences.

Deferred Shading

- Screen-space shading method
- Decouples geometry from lighting
- Metal Gear Solid example

Lighting

- Global (ambient, sky)
- Point light (light bulb)
- Directional light (sun rays)
- Spot light (flashlight)

Physics-based Rendering

- Attempt to imitate how light flow works in real world.
- May include: albedo, gloss, reflection, diffusion, metal

Activity

1. Pick a game you've played recently
2. Recall what graphics settings the game offered
3. Explain in own words what each setting does

We will consider some of the common ones.

Ambient Occlusion

- Shadows created by objects blocking ambient light
- Screen Space Ambient Occlusion (SSAO) method (by Crytek) uses only the depth (Z) buffer

Post-processing

- Bloom, Lens Flare, Vignette
- Blending: example

RGBA colors and Blending

- Values range: memory implications and multithreaded drawing
- Alpha opacity / transparency

Anti-aliasing

- Smoothing of “aliased” (jagged) lines
- FXAA
- MSAA

Filtering

- Enhances (e.g. reduces blur) texture quality when viewed at various angles
- Bilinear
- Trilinear
- Anisotropic

Particle Effects

- A system that controls large numbers of particles.
- A particle is a small geometric (possibly textured) object with certain properties, such as velocity, acceleration, scale, etc.

Particle Effects Impl

Let's implement a simple particle subsystem!

Interpolators

- Known as easing / tweening
- Affect the rate of change
- Examples include: linear, exponential, elastic, etc.

Interpolators Theory

- Given a time value in range `[0..1]`, an interpolator converts it to a progress value `[0..1]`.
- Essentially, can be represented as a function
- Example

Let's consider the theory in detail using a simple example.

Activity

1. In fxgl-animation module, identify various `interpolator` implementations.
2. Implement (e.g. on paper) your own custom interpolator (function).

UI / HUD

- Drawn in the orthographic view (typically)
- Provides information about the game / player state
- Includes aesthetically pleasing visual effects (animations using interpolators)

Activity:

- Design and implement a simple UI button.
- The button, when clicked, should perform *some* action.
- (extra) Draw some text for the button.
- (extra) Pressed / unpressed modes, so the user sees when clicked.
- (extra) Animate the button using your interpolator.

Audio

Sounds and music are important to provide the atmosphere.

Common Audio Formats

- uncompressed: wav, aiff
- lossless: flac
- lossy: mp3, ogg

Encoding (audio and other)

Typically (not just audio)

1. Header data (e.g. version, bit rate, frequency, etc.)
2. Metadata (e.g. owner, date, etc.)
3. Raw or structured data

Audio Formats SDL

- Provided by SDL2_mixer
- WAV, FLAC, MikMod MOD, Timidity MIDI, Ogg Vorbis, and SMPEG MP3

SDL2_mixer usage

```
// note use of raw pointers in SDL
Mix_Chunk* sound = Mix_LoadWAV(WAV_PATH);

// what are -1 and 0? use SDL2_mixer documentation
Mix_PlayChannel(-1, sound, 0);

// we like to clean up after ourselves
Mix_FreeChunk(sound);
```

Activity

Implement - play a sound file in the provided code base.

Positional Sound

Using balance and volume of the speakers you can play positional sound.

Positional Sound Theory

1. Identify the direction
2. Compute the distance
3. What's next?

Conclusion

- Graphics not complex if you use high-level abstractions
- Audio is important and not difficult to implement using SDL